

Apache MXNet 框架的新接口 – Gluon

符号 (Symbolic)

- 效率&可移植性
- 但是难于使用



Gluon

- 命令用于开发
- 符号用于部署



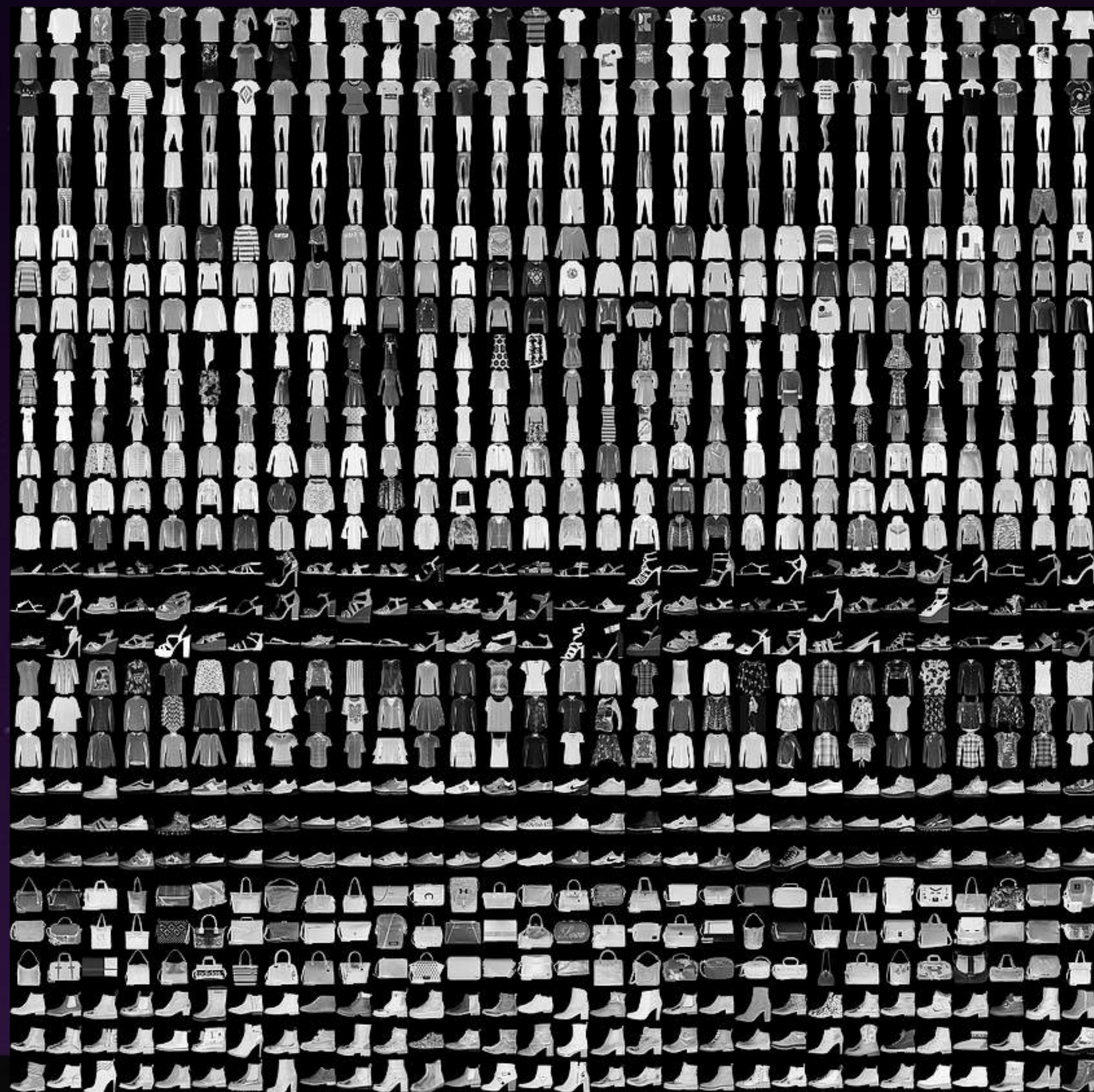
命令 (Imperative)

- 灵活
- 性能或许不够好



Apache MXNet Gluon 演示 – Fashion-MNIST

- 是一个替代MNIST手写数字集的图像数据集
- 涵盖了10种类别的共7万个不同商品的图片
- 60000/10000的训练测试数据划分，28x28的灰度图片
- 项目地址：
<https://github.com/zalandoresearch/fashion-mnist/blob/master/README.zh-CN.md>



Apache MXNet Gluon 演示 – Fashion-MNIST

jupyter fashion_mnist Last Checkpoint: Last Monday at 5:46 PM (autosaved)

FileEditViewInsertCellKernelHelp

Trustedmy-mxnet

+

3<

↑

↓

Run

C

Markdown

Logout

关于 Fashion mnist

FashionMNIST 是一个替代 MNIST 手写数字集 [1] 的图像数据集。它是由 Zalando（一家德国的时尚科技公司）旗下的研究部门提供。其涵盖了来自 10 种类别的共 7 万个不同商品的正面图片。

FashionMNIST 的大小、格式和训练集/测试集划分与原始的 MNIST 完全一致。60000/10000 的训练测试数据划分，28x28 的灰度图片。你可以直接用它来测试你的机器学习和深度学习算法性能，且不需要改动任何的代码。Fashion-MNIST 的目的是要成为 MNIST 数据集的一个直接替代品。作为算法作者，你不需要修改任何的代码，就可以直接使用这个数据集。Fashion-MNIST 的图片大小，训练、测试样本数及类别数与经典 MNIST 完全相同。

类别标注

每个训练和测试样本都按照以下类别进行了标注：

标注编号	描述
0	T-shirt/top (T恤)
1	Trouser (裤子)
2	Pullover (套衫)
3	Dress (裙子)
4	Coat (外套)
5	Sandal (凉鞋)

Apache MXNet 框架 – AWS 的基础设施

AWS EC2 实例

CPU 优化实例

- C4.8xlarge (36 threads, 60GB RAM, 4Gbit)
- M4.16xlarge (64 threads, 256GB RAM, 10Gbit)

GPU 实例

- P2.16xlarge (16X NVIDIA Kepler K80, 64 threads, 732GB RAM, 20Gbit)
- G3.16xlarge (4XNVIDIA Maxwell M60, 64 threads, 488GB RAM, 20Gbit)
- NVIDIA Volta 实例即将发布



Apache MXNet 框架 – AWS 的基础设施

AWS Machine Image for Deep Learning



Deep Learning AMI Amazon Linux Version

★★★★★ (11) | 2.3_Jun2017 [Previous versions](#) | Sold by [Amazon Web Services](#)

\$0.0059 to \$24.672/hr incl EC2 charges + other AWS usage fees

Linux/Unix, Amazon Linux 2016.09 | 64-bit Amazon Machine Image (AMI) | Updated: 6/30/17

The Deep Learning AMI is a base Amazon Linux image provided by Amazon Web Services for use on Amazon Elastic Compute Cloud(Amazon EC2).It is designed to provide a stable, secure, ...

[More info](#)

Security Patches(for Stackguard vulnerability) applied on the base Amazon Linux AMI

MXNet compiled with S3 support

7 Deep Learning Frameworks - contains the most popular Deep Learning Frameworks(MXNet, Caffe, Caffe2, Tensorflow, Theano, Torch and CNTK)



Deep Learning AMI Ubuntu Version

★★★★★ (6) | 2.2_Aug2017 [Previous versions](#) | Sold by [Amazon Web Services](#)

\$0.0059 to \$24.672/hr incl EC2 charges + other AWS usage fees

Linux/Unix, Ubuntu 16.04 | 64-bit Amazon Machine Image (AMI) | Updated: 8/15/17

The Deep Learning AMI is a base Ubuntu image provided by Amazon Web Services for use on Amazon Elastic Compute Cloud(Amazon EC2).It is designed to provide a stable, secure, and ...

[More info](#)

Used Ubuntu 16.04 (ami-d15a75c7) as the base AMI with CUDA 8 support

Framework upgrades for MXNet(0.10.0.post1), Tensorflow(v1.2.0), Caffe2(v0.7.0), Caffe(1.0), CNTK(v2.0), Theano(rel-0.9.0)

7 Deep Learning Frameworks - contains the most popular Deep Learning Frameworks(MXNet, Caffe, Caffe2, Tensorflow, Theano, Torch and CNTK)



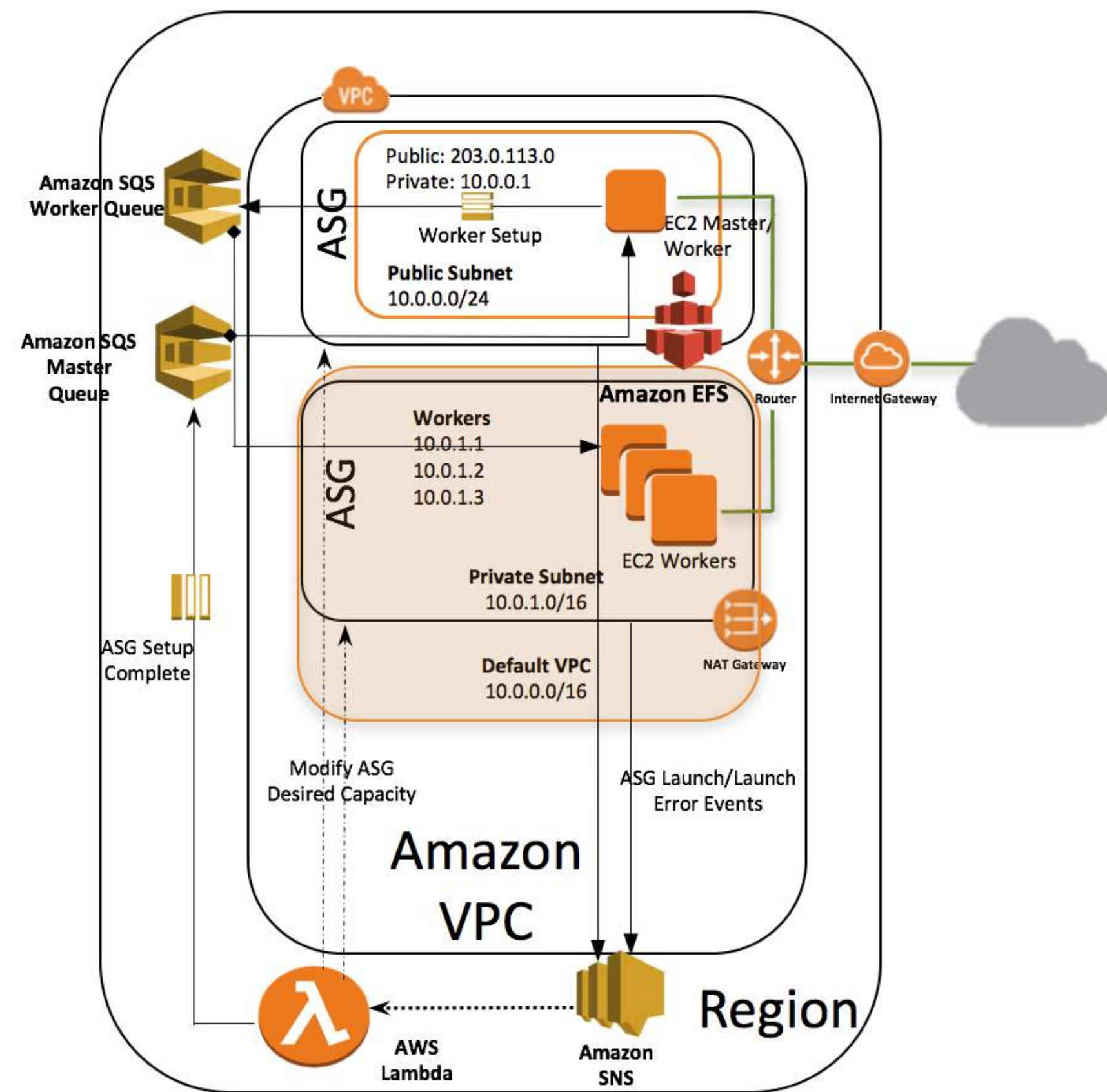
Apache MXNet 框架 – AWS 的基础设施

AWS CloudFormation Template for Deep Learning

项目地址: <https://github.com/aws-labs/deeplearning-cfn>



AWS Deep Learning Cluster



Apache MXNet 框架演示 – AWS上的集群图像分类训练



CIFAR-10 数据集包含了6万个32x32的彩色图片，共有10个类别每个类别有 6,000 张图片。其中，5万张训练图片以及1万张测试图片。

```
$ssh -A -i xxx.pem ubuntu@xxx.xxx.xxx.xxx
$mkdir $EFS_MOUNT/cifar_model/

$cd $EFS_MOUNT/deeplearning-cfn/examples/mxnet/example/image
classification/

$ ../../tools/launch.py -n $DEEPLARNING_WORKERS_COUNT \
-H $DEEPLARNING_WORKERS_PATH python train_cifar10.py \
--network resnet --num-layers 110 --kv-store dist_device_sync \
--model-prefix $EFS_MOUNT/cifar_model/cifar --num-epochs 300 --batch-size 128
```


Apache MXNet 框架演示 – AWS上的集群图像分类训练

```
1 |#!/usr/bin/env bash
2 |
3 |sudo nvidia-smi -pm 1
4 |#Set the persistence mode for the target GPUs.
5 |sudo nvidia-smi -acp 0
6 |#Toggle whether applications clocks can be changed by all users or only by root
7 |sudo nvidia-smi --auto-boost-permission=0
8 |# Allow non-admin/root control over auto boost mode.
9 |sudo nvidia-smi -ac 2505,875
10|#Specifies maximum <memory,graphics> clocks as a pair (e.g. 2000,800) that defines GPU's speed while running applications on a GPU.
11|
12|#In case of Intel® Xeon® Processors
13|export OMP_NUM_THREADS=$((($(grep 'core id' /proc/cpuinfo | sort -u | wc -l)*2))
14|
15|export MXNET_ENABLE_GPU_P2P=0
16|export MXNET_CPU_WORKER_NTHREADS=32
17|
18|if [ ! -d $EFS_MOUNT/cifar_model ]; then
19|    mkdir $EFS_MOUNT/cifar_model/
20|fi
21|
22|cd $EFS_MOUNT/deeplearning-cfn/mxnet/example/image-classification/
23|
24|dt=$(date +%d/%m/%Y %H:%M:%S)
25|echo "Start processing ..."
26|echo "Start processing at $dt" >> result.txt
27|
28|START=$(date +%s)
29|
30|##CPU
31|#./../tools/launch.py -n $DEEPLARNING_WORKERS_COUNT \
32|#-H $DEEPLARNING_WORKERS_PATH python3 train_cifar10.py \
33|#--network resnet --num-layers 50 --kv-store dist_device_sync \
34|#--model-prefix $EFS_MOUNT/cifar_model/cifar --num-epochs 10
35|
36|##GPU
37|../tools/launch.py -n $DEEPLARNING_WORKERS_COUNT -H $DEEPLARNING_WORKERS_PATH \
38|python train_cifar10.py --gpus $(seq -s , 0 1 $((($DEEPLARNING_WORKER_GPU_COUNT - 1))) \
39|--network resnet --num-layers 110 --kv-store dist_device_sync \
40|--model-prefix $EFS_MOUNT/cifar_model/cifar --num-epochs 300 --batch-size 128
41|
42|
43|END=$(date +%s)
44|dt=$(date +%d/%m/%Y %H:%M:%S);
45|echo "Finished at $dt" >> result.txt
46|
47|DIFF=$(( $END - $START ))
48|
49|echo "It took $DIFF seconds"
50|echo "It took $DIFF seconds" >> result.txt
51|echo "Done!"
```

性能对比：

计算环境	1 x P2.16xLarge	5 x P2.16xlarge
运行时间	36,122s = 10.3h	7,632s = 2.12h

效率提升 473%



Apache MXNet 框架 – 从源代码编译安装

```
USE_CUDA=1
```

```
USE_CUDNN=1
```

```
USE_OPENMP=1
```

```
USE_CUDA_PATH=/usr/local/cuda
```

```
USE_BLAS=mkl
```

```
USE_MKL2017=1
```

```
USE_MKL2017_EXPERIMENTAL=1
```

```
MKLML_ROOT=/usr/local
```

```
USE_INTEL_PATH=/opt/intel
```

```
USE_OPENCV=1
```

```
USE_S3=1
```

```
USE_NVRTC=1
```

```
USE_DIST_KVSTORE=1
```

mxnet/make/config.mk

```
##K80 3.7,M60 5.2, GTX 1070 6.1
```

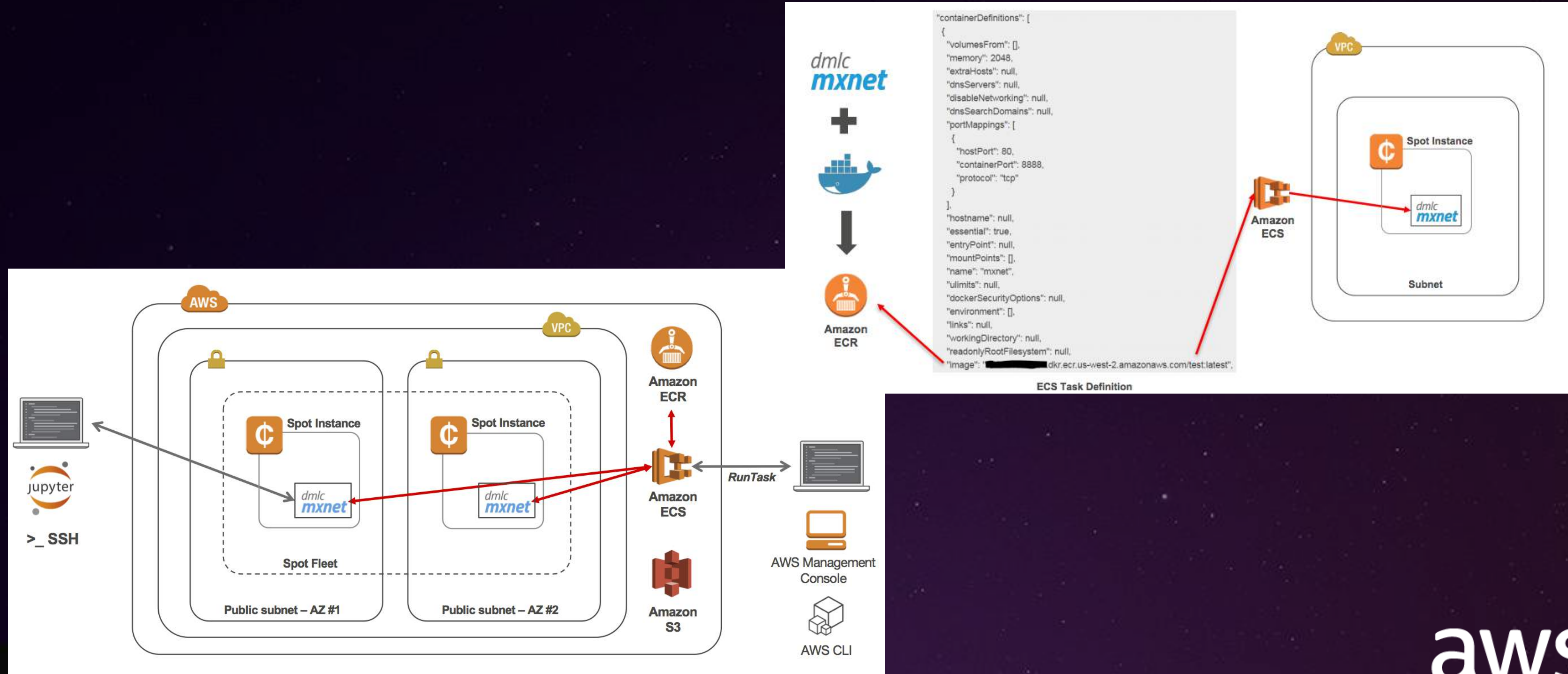
```
MSHADOW_NVCCFLAGS += -gencode
```

```
arch=compute_61,code=sm_61
```

mxnet/mshadow/make/mshadow.mk



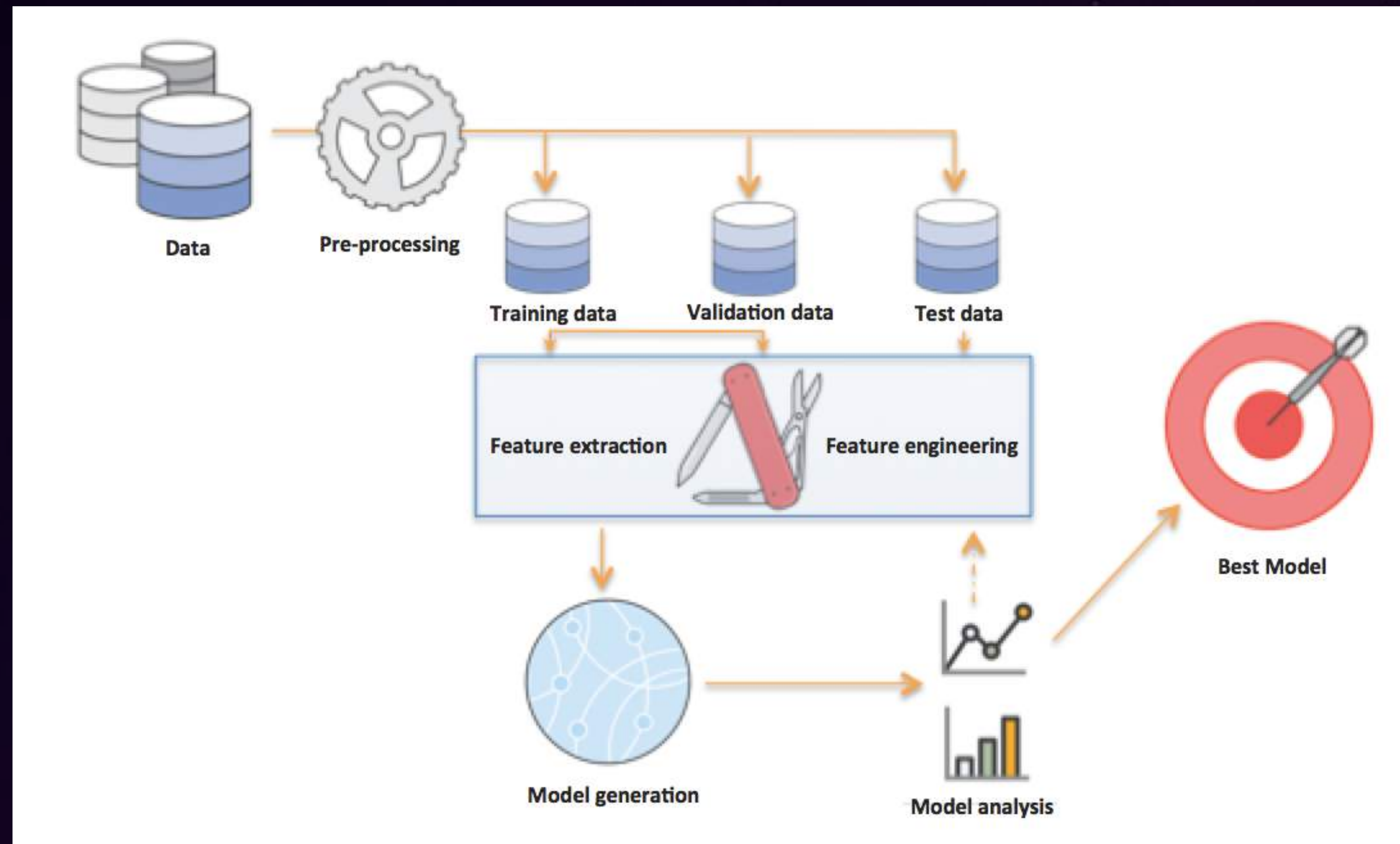
在Amazon ECS容器服务上部署Apache MXNet



来源 : <https://github.com/aws-labs/ecs-deep-learning-workshop>



使用AWS Lambda和MXNet 进行预测



参考：

- [awslabs/mxnet-lambda](https://aws-labs.github.io/mxnet-lambda/)
- [Seamlessly Scale Predictions with AWS Lambda and MXNet](#)

Apache MXNet @ Amazon

- Applied Research
 - Core Research
 - Alexa
 - Demand Forecasting
 - Risk Analytics
 - Search
 - Recommendation
 - AI Services | Rek, Lex, Polly
- Q&A Systems
 - Supply Chain Optimization
 - Advertising
 - Machine Translation
 - Video Content Analysis
 - Robotics
 - Lots of Computer Vision...
 - Lots of NLP/U ...

Apache MXNet 参考资源

- MXNet 官网 , <http://mxnet.ai>
- MXNet 项目代码 , <https://github.com/apache/incubator-mxnet>
- Deep Learning - The Straight Dope, <https://github.com/zackchase/mxnet-the-straight-dope>
- 动手学深度学习, <https://zh.gluon.ai>
- AWS AI Blog, <https://aws.amazon.com/cn/blogs/ai/>
- Apache MXNet on AWS, <https://aws.amazon.com/cn/mxnet/>
- AWS Labs, <https://github.com/aws-labs>



关注QCon微信公众号
获得更多干货!

Thanks!

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方: **Geekbang** > **InfoQ**
极客邦科技